

REPORT DOCUMENTATION PAGE			Form Approved OMB NO. 0704-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 09-03-2016		2. REPORT TYPE Conference Proceeding		3. DATES COVERED (From - To) -	
4. TITLE AND SUBTITLE MI-ANFIS: A Multiple Instance AdaptiveNeuro-Fuzzy Inference System			5a. CONTRACT NUMBER W911NF-14-1-0589		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHORS Amine B. Khalifa , Hichem Frigui			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Louisville 2301 S. Third Street Jouett Hall Louisville, KY 40208 -1838			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSOR/MONITOR'S ACRONYM(S) ARO		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) 66411-CS.3		
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT We introduce a novel adaptive neuro-fuzzy architecture based on the framework of Multiple Instance Fuzzy Inference. The new architecture called Multiple Instance-ANFIS (MI-ANFIS), is an extension of the standard Adaptive Neuro Fuzzy Inference System (ANFIS) [1] that is designed to handle reasoning with multiple instances (bags of instances) as input and capable of learning from ambiguously labeled data. In multiple instance problems the training data is ambiguously labeled. Instances are grouped into bags, labels of bags are known but not those of individual instances. Multiple Instance Learning (MIL)					
15. SUBJECT TERMS Multiple Instance Learning; Fuzzy Inference					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Hichem Frigui
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU			19b. TELEPHONE NUMBER 502-852-2009

Report Title

MI-ANFIS: A Multiple Instance Adaptive Neuro-Fuzzy Inference System

ABSTRACT

We introduce a novel adaptive neuro-fuzzy architecture based on the framework of Multiple Instance Fuzzy Inference. The new architecture called Multiple Instance-ANFIS (MI-ANFIS), is an extension of the standard Adaptive Neuro Fuzzy Inference System (ANFIS) [1] that is designed to handle reasoning with multiple instances (bags of instances) as input and capable of learning from ambiguously labeled data. In multiple instance problems the training data is ambiguously labeled. Instances are grouped into bags, labels of bags are known but not those of individual instances. Multiple Instance Learning (MIL) deals with learning a classifier at the bag level. Over the years many solutions to this problem have been proposed. However, no MIL formulation employing fuzzy inference exists in the literature. In this paper, we develop MI-ANFIS that generalizes ANFIS inference systems to account for ambiguity and reason with multiple instances. We also develop a learning algorithm to learn the parameters of MI-ANFIS. The proposed MI-ANFIS is tested and validated using a synthetic and benchmark data sets suitable for MIL problems.

Conference Name: IEEE Conference on Fuzzy Systems

Conference Date: August 02, 2015

MI-ANFIS: A Multiple Instance Adaptive Neuro-Fuzzy Inference System

Amine B. Khalifa
Multimedia Research Lab
CECS Department
University of Louisville
Louisville, KY 40292, USA
Email: a0benk01@louisville.edu

Hichem Frigui
Multimedia Research Lab
CECS Department
University of Louisville
Louisville, KY 40292, USA
Email: h.frigui@louisville.edu

Abstract—We introduce a novel adaptive neuro-fuzzy architecture based on the framework of Multiple Instance Fuzzy Inference. The new architecture called Multiple Instance-ANFIS (MI-ANFIS), is an extension of the standard Adaptive Neuro Fuzzy Inference System (ANFIS) [1] that is designed to handle reasoning with multiple instances (bags of instances) as input and capable of learning from ambiguously labeled data. In multiple instance problems the training data is ambiguously labeled. Instances are grouped into bags, labels of bags are known but not those of individual instances. Multiple Instance Learning (MIL) deals with learning a classifier at the bag level. Over the years many solutions to this problem have been proposed. However, no MIL formulation employing fuzzy inference exists in the literature. In this paper, we develop MI-ANFIS that generalizes ANFIS inference systems to account for ambiguity and reason with multiple instances. We also develop a learning algorithm to learn the parameters of MI-ANFIS. The proposed MI-ANFIS is tested and validated using a synthetic and benchmark data sets suitable for MIL problems.

I. INTRODUCTION

THE standard Adaptive Neuro-Fuzzy Inference System (ANFIS)[1] is a universal approximator that combines the learning and modeling power of neural networks and fuzzy logic into an adaptive inference system. Neural networks deal with imprecise data by training, while fuzzy logic can deal with the uncertainty of human cognition [2]. ANFIS offers an alternative to rules' identification. While Mamdani [3] and Sugeno [4] fuzzy systems identify rules based on intuition, ANFIS, in contrast, jointly learns the optimal input space partition and the optimal output parameters through optimization. ANFIS is considered a hybrid intelligent system and it provides a systematic approach to learn fuzzy rules from a given input-output dataset using supervised learning. Typically, in supervised learning problems, access to large labeled training datasets improves the performance of the devised algorithms by overcoming noise and adding robustness and generalization to unseen examples. Even though, large amounts of data are available and could be used for learning, in many applications, this data is typically labeled ambiguously and at a coarse level. In fact, labels, or tags, tend to be associated with collections of samples rather than single samples. For example, in image annotation, tags could be used as indicators of the existence of objects of interests within the images (sky, sea, beach,...). However, the exact location of those objects is not available and is too tedious to extract for large collection

of images. An alternative and a relatively new framework of learning that tackles the inherent ambiguity better than supervised learning, is the Multiple Instance Learning (MIL) paradigm [5]. Unlike standard supervised learning, in MIL, an example is not a simple data point, but a collection of instances, called a bag. Each bag can contain a different number of instances. A bag is labeled negative if all of its instances are negative, and positive if at least one of its instances is positive¹. Positive bags can encode ambiguity since the instances themselves are not labeled. Given a training set of labeled bags, the goal of MIL is to learn a concept that predicts the labels of training data and generalizes to predict the labels of testing bags [6].

To effectively take full advantage of the standard ANFIS system in the context of MIL, bags need to be labeled at the instances level by human experts to make learning possible [7]. Unfortunately, this process is tedious, ambiguous, subjective, and prone to errors. To address this major limitation, we introduce an adaptive neuro-fuzzy architecture that is designed to handle reasoning with bags of instances as input and capable of learning from ambiguously labeled data. The new architecture is called Multiple Instance-ANFIS (MI-ANFIS). The rest of this paper is organized as follows. Section II describes the architecture of the proposed MI-ANFIS, and a corresponding learning algorithm is introduced in Section III. Section IV presents the experimental results on a synthetic and benchmark data sets. Finally, we provide the conclusions in section V.

II. MI-ANFIS ARCHITECTURE

In the following, let B_p be a bag of M_p instances with the j th instance denoted as \mathbf{x}_{pj} . \mathbf{x}_{pj} is in turn a D dimensional vector with elements x_{pjk} corresponding to features, i.e.,

$$B_p = \begin{pmatrix} x_{p11} & x_{p12} & \dots & x_{p1D} \\ x_{p21} & x_{p22} & \dots & x_{p2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{pM_p1} & x_{pM_p2} & \dots & x_{pM_pD} \end{pmatrix}. \quad (2)$$

Note that the number of instances can vary between bags (M_p depends on B_p). A bag is labeled positive if at least one of its instances is positive, and negative if all of its instances are negative.

¹Note that positive bags may also contain negative instances.

$$\begin{aligned} \mathcal{R}^1(\mathbf{B}_p) : \bigvee_{j=1}^{M_p} (\text{If } x_{pj1} \text{ is } A_{11} \text{ and } \dots \text{ and } x_{pjD} \text{ is } A_{1D}), \text{ then } f_1 = C(\mathbf{x}_{p1} \cdot \mathbf{b}^1, \mathbf{x}_{p2} \cdot \mathbf{b}^1, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^1) \\ \mathcal{R}^2(\mathbf{B}_p) : \bigvee_{j=1}^{M_p} (\text{If } x_{pj1} \text{ is } A_{21} \text{ and } \dots \text{ and } x_{pjD} \text{ is } A_{2D}), \text{ then } f_2 = C(\mathbf{x}_{p1} \cdot \mathbf{b}^2, \mathbf{x}_{p2} \cdot \mathbf{b}^2, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^2) \end{aligned} \quad (1)$$

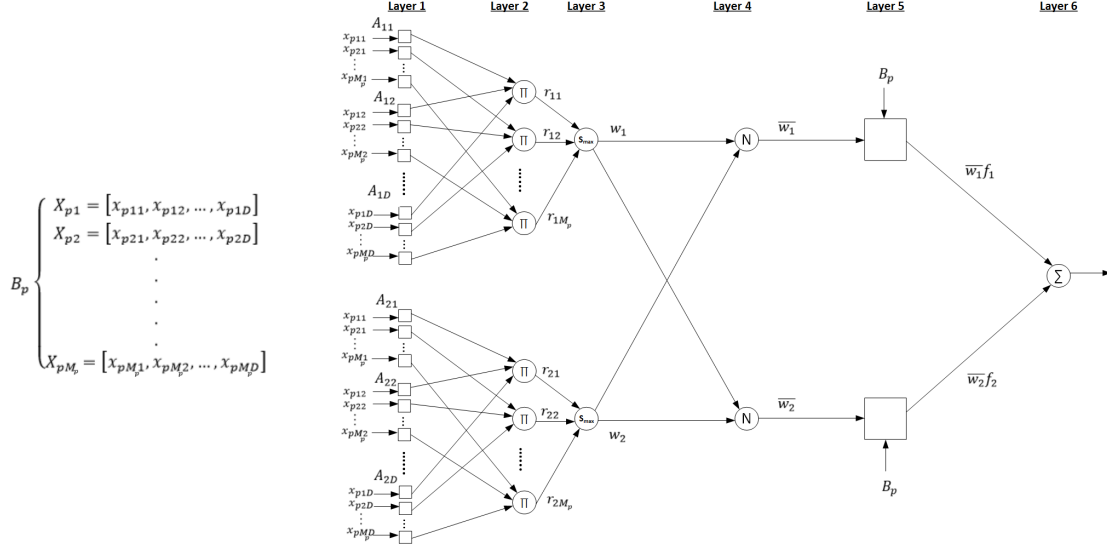


Fig. 1. Architecture of the proposed multiple instance Adaptive Neuro-Fuzzy Inference System

We introduce our MI-ANFIS for the simple case of two rules. Equation (1) describes the MI-ANFIS with two Sugeno rules. Here, A_k^i is a fuzzy set associated with the k th instance feature, and “ \bigvee ” is a joint operator that can be any T-conorm (or, max, sum, etc.). $\mathbf{b}^i = b_0^i, \dots, b_D^i$ is a set of polynomial coefficients. The premise part of the rule is evaluated as in the ANFIS case. To evaluate the consequent part, first the linear response of each instance is computed, i.e., $\mathbf{x}_{pj} \cdot \mathbf{b}^i$. Then, a function C is used to compute the final output by combining the instances’ responses. Many functions could be used and the choice should be domain-specific. For instance, the “max” function has been used in many applications.

Figure 1 illustrates the proposed MI-ANFIS architecture, the upper part and lower part of the network correspond to the first and second fuzzy rules. As in the traditional ANFIS, nodes at the same layer have similar functions. We denote the output of the i th node in layer l as $O_{l,i}$

Layer 1 is an adaptive layer, it calculates the degree to which a given input instance satisfies a quantifier A . Every node evaluates the membership degree of an input instance in the fuzzy set $A_{k,j}$ of membership function $\mu_{A_{k,j}}$. Generally, $\mu_{A_{k,j}}$ is a parameterized membership function (MF), for example a Gaussian MF with

$$\mu_{A_{k,j}}(x) = \exp\left(\frac{-(x - c_{kj})^2}{2\sigma_{kj}^2}\right). \quad (3)$$

In (3), c_{kj} and σ_{kj} are the mean and variance of the gaussian function, and are referred to as the **premise parameters**.

Layer 2 is a fixed layer, every node computes the product of all incoming inputs. In the context of multiple instance fuzzy logic, layer 2 evaluates the degree of truth of proposition instances, or simply, “truth instances”. The output of this layer is

$$O_{2,i} = r_{\lceil i/M_p \rceil, i[M_p]} = \prod_{j=1}^D \mu_{A_{\lceil i/M_p \rceil, j}}(x_{p, i[M_p], j}), \quad (4)$$

where $\lceil \cdot \rceil$ is a ceiling operator, and $i[M_p]$ is $i \bmod M_p$. As in the traditional ANFIS, any T-norm can be used as the node function in this layer.

Layer 3 is a new addition when compared to the traditional ANFIS. Every node aggregates the truth instances of the previous layer by means of a smooth T-conorm. In this paper, we use a smooth approximation of the “max” T-conorm known as the “softmax” function (\mathcal{S}_α):

$$\begin{aligned} softmax_\alpha(x_1, x_2, \dots, x_n) = \\ \mathcal{S}_\alpha(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \frac{x_i \cdot e^{\alpha x_i}}{\sum_{j=1}^n e^{\alpha x_j}}. \end{aligned} \quad (5)$$

In (5), as pointed by Maron in [8], the parameter α determines how closely softmax approximates the max operator. As α approaches ∞ , softmax’s behavior approaches max. When $\alpha = 0$, it calculates the mean. As α approaches $-\infty$, softmax’s behavior approaches the min operator.

The outputs of this layer are the firing strength of

the multiple instance fuzzy rules defined by layers 1 through layer 3. i.e.,

$$O_{3,i} = w_i = \mathcal{S}_\alpha(\{r_{i,j}\}_{j=1}^{M_p}), \quad (6)$$

where α is a fixed constant. Layer 3 is also a fixed layer.

Layer 4 is a fixed layer. Every node labeled N of this layer calculates the **normalized firing strength** of each rule:

$$O_{4,i} = \bar{w}_i = \frac{w_i}{\sum_{j=1}^{|O_3|} w_j}, \quad (7)$$

where $|O_3|$ is the number of rules.

Layer 5 is an adaptive layer. Every node i in this layer computes the output of the i 'th multiple instance rule. Because our MI-ANFIS is functionally equivalent to the a multiple instance Sugeno fuzzy inference system, the output of each rule will be computed using the combining function

$$O_{5,i} = C(\mathbf{x}_{p1} \cdot \mathbf{b}^i, \mathbf{x}_{p2} \cdot \mathbf{b}^i, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^i), \quad (8)$$

where $\mathbf{x}_{pj} = \{x_{p,j,1}, \dots, x_{p,j,D}\}$ for $j = 1, \dots, M_p$, and $\mathbf{b}^i = \{b_0^i, \dots, b_D^i\}$ is a set of polynomial coefficients. The parameters $\{\mathbf{b}^i\}_{i=1}^{|O_3|}$ are referred to as the **consequents parameters**. The only constraint on C is it has to be a smooth function to allow for optimization techniques to be applied. In the following, we choose “softmax” as the combining function for this layer. In this case (8) is equivalent to:

$$O_{5,i} = \bar{w}_i \mathcal{S}_\alpha(\mathbf{x}_{p1} \cdot \mathbf{b}^i, \mathbf{x}_{p2} \cdot \mathbf{b}^i, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^i), \quad (9)$$

note that the constant α here is not necessary the same as in Layer 3.

Layer 6 is a fixed layer with a single node labeled Σ . As in the traditional ANFIS, it computes the overall output of the system using

$$O_{6,1} = \sum_{i=1}^{|O_3|} O_{5,i} = \sum_{i=1}^{|O_3|} \bar{w}_i \mathcal{S}_\alpha(\mathbf{x}_{p1} \cdot \mathbf{b}^i, \mathbf{x}_{p2} \cdot \mathbf{b}^i, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^i). \quad (10)$$

III. BASIC LEARNING ALGORITHM

To identify the parameters of the proposed MI-ANFIS network, we propose a variation of the basic learning algorithm presented by Jang [9]. Our variation is different from the ANFIS standard backpropagation learning rule due to the additional layers our network introduced, as well as the use of new activation functions at the nodes level, such as “softmax”.

A. BackPropagation Learning Rule

In the following, we assume that we have N training bags, $\mathcal{B} = \{\mathbf{B}_p \mid p = 1, \dots, N\}$, and their corresponding labels $\mathcal{T} = \{t_p \mid p = 1, \dots, N\}$.

First, for the p th training bag, we compute a squared error

measure commonly used in the backpropagation algorithm and defined as

$$E_p = (t_p - O_p)^2, \quad (11)$$

In (11), t_p is the desired bag output, and O_p is the computed output of the network when presented with training bag p . Equation (11) demonstrates the need for MI-ANFIS. In fact, due to the absence of instances' labels, errors can be computed only at the bag level. Errors at the instance level cannot be computed and are not needed as we will show later. The overall error measure of the network is

$$E = \sum_{p=1}^N E_p. \quad (12)$$

To develop the gradient descent optimization on E , we compute the error rate for the p th training bag and for each node output $O_{l,i}$. This error rate $\varepsilon_{l,i}$ ($1 \leq l \leq 6$ indicates the MI-ANFIS layer) is defined as

$$\varepsilon_{l,i} = \frac{\partial E_p}{\partial O_{l,i}}. \quad (13)$$

The error rate at the output node is given as following

$$\varepsilon_{6,1} = \frac{\partial E_p}{\partial O_{6,1}} = \frac{\partial E_p}{\partial O_p} = -2(t_p - O_p). \quad (14)$$

For non-output nodes (i.e. internal nodes, $l < 6$), we derive the error rate using the chain rule

$$\varepsilon_{l,i} = \frac{\partial E_p}{\partial O_{l,i}} = \sum_{h=1}^{Card(l+1)} \frac{\partial E_p}{\partial O_{l+1,h}} \frac{\partial O_{l+1,h}}{\partial O_{l,i}}, \quad (15)$$

where $Card(l+1)$ refers the number of nodes at layer $l+1$. Next, we seek to minimize the network error with respect to the premise parameters $\{c_{kj}, \sigma_{kj} \mid 1 \leq k \leq |O_3|, 1 \leq j \leq D\}$, and with respect to consequents parameters $\{\mathbf{b}^i\}_{i=1}^{|O_3|}$. The error rate with respect to a generic parameter θ can be computed using

$$\frac{\partial E_p}{\partial \theta} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \theta}, \quad (16)$$

where S is the set of nodes whose outputs depend on θ . Using (12), the total error rate is given by

$$\frac{\partial E}{\partial \theta} = \sum_{p=1}^N \frac{\partial E_p}{\partial \theta}. \quad (17)$$

1) *Update Rule For Premise Parameters*: First we compute the error rate for the premise parameters c_{kj} and σ_{kj} . We have

$$\frac{\partial E_p}{\partial c_{kj}} = \sum_{i=1}^{M_p} \frac{\partial E_p}{\partial O_{(1,i+[(k-1)D+(j-1)]M_p)}} \frac{\partial O_{(1,i+[(k-1)D+(j-1)]M_p)}}{\partial c_{kj}}. \quad (18)$$

and,

$$\frac{\partial E_p}{\partial \sigma_{kj}} = \sum_{i=1}^{M_p} \frac{\partial E_p}{\partial O_{(1,i+[(k-1)D+(j-1)]M_p)}} \frac{\partial O_{(1,i+[(k-1)D+(j-1)]M_p)}}{\partial \sigma_{kj}}. \quad (19)$$

Using the chain rule defined in (15), it can be shown that

$$\begin{aligned} \frac{\partial E_p}{\partial c_{kj}} = & -2(t_p - O_p) \times \mathcal{S}_\alpha(\mathbf{x}_{p1} \cdot \mathbf{b}^k, \mathbf{x}_{p2} \cdot \mathbf{b}^k, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^k) \times \frac{\sum_{l=1}^{|O_3|} w_l - w_k}{\left(\sum_{l=1}^{|O_3|} w_l\right)^2} \quad \text{2) Update Rule For Consequents Parameters:} \\ & \text{The error rate for the consequent parameters } \{\mathbf{b}^i = \{b_0^i, \dots, b_D^i\}, i = 1 \dots |O_3|\} \text{ is defined as} \\ & \frac{\partial E_p}{\partial \mathbf{b}^i} = \left(\frac{\partial E_p}{\partial b_0^i}, \frac{\partial E_p}{\partial b_1^i}, \dots, \frac{\partial E_p}{\partial b_D^i} \right). \quad (24) \\ & \times \sum_{i=1}^{M_p} \left[\frac{e^{\alpha r_{k,(i+(k-1)M_p)}}}{\sum_{m=1}^{M_p} e^{\alpha r_{k,m}}} \left[1 + \alpha \left(r_{k,(i+(k-1)M_p)} - \mathcal{S}_\alpha(\{r_{k,m}\}_{m=1}^{M_p}) \right) \right] \right. \\ & \times \prod_{d=1, d \neq j}^D \mu_A \left(\left[\frac{(i+(k-1)M_p)}{M_p} \right]_{,d} \right) \left(x_{p,(i+(k-1)M_p)[M_p],d} \right) \\ & \times \frac{(x_{p,(i+(k-1)M_p)[M_p],j} - c_{kj})}{\sigma_{kj}^2} \\ & \left. \times \exp\left(-\frac{(x_{p,(i+(k-1)M_p)[M_p],j} - c_{kj})^2}{2\sigma_{kj}^2}\right) \right]. \quad (20) \end{aligned}$$

As in the standard ANFIS, an update formula for the parameter c_{kj} is given by

$$\Delta c_{kj} = -\eta \frac{\partial E}{\partial c_{kj}}, \quad (21)$$

where η is a learning rate determined in a similar manner to that of standard backpropagation algorithm [9].

The update formula for σ_{kj} can be derived in a similar manner. It can be shown that

$$\begin{aligned} \frac{\partial E_p}{\partial \sigma_{kj}} = & -2(t_p - O_p) \times \mathcal{S}_\alpha(\mathbf{x}_{p1} \cdot \mathbf{b}^k, \mathbf{x}_{p2} \cdot \mathbf{b}^k, \dots, \mathbf{x}_{pM_p} \cdot \mathbf{b}^k) \times \frac{\sum_{l=1}^{|O_3|} w_l - w_k}{\left(\sum_{l=1}^{|O_3|} w_l\right)^2} \\ & \times \sum_{i=1}^{M_p} \left[\frac{e^{\alpha r_{k,(i+(k-1)M_p)}}}{\sum_{m=1}^{M_p} e^{\alpha r_{k,m}}} \left[1 + \alpha \left(r_{k,(i+(k-1)M_p)} - \mathcal{S}_\alpha(\{r_{k,m}\}_{m=1}^{M_p}) \right) \right] \right. \\ & \times \prod_{d=1, d \neq j}^D \mu_A \left(\left[\frac{(i+(k-1)M_p)}{M_p} \right]_{,d} \right) \left(x_{p,(i+(k-1)M_p)[M_p],d} \right) \\ & \times \frac{(x_{p,(i+(k-1)M_p)[M_p],j} - c_{kj})^2}{\sigma_{kj}^3} \\ & \left. \times \exp\left(-\frac{(x_{p,(i+(k-1)M_p)[M_p],j} - c_{kj})^2}{2\sigma_{kj}^2}\right) \right]. \quad (22) \end{aligned}$$

And the update formula for σ_{kj} is as follows

$$\Delta \sigma_{kj} = -\eta \frac{\partial E}{\partial \sigma_{kj}}, \quad (23)$$

where η is the same learning rate as in (21)

Equations (21) and (23) can be used to update c_{kj} and σ_{kj} parameters either on-line, bag by bag (we want to emphasize here that the on-line learning is not achieved instance by instance, but rather bag by bag), or after presentation of the entire data set. This latter mode of learning is known as batch-learning or off-line learning. Next, we develop the update rules for the consequents parameters.

where,

$$\frac{\partial E_p}{\partial b_j^i} = \frac{\partial E_p}{\partial O_{(5,i)}} \frac{\partial O_{(5,i)}}{\partial b_j^i}, \text{ for } j = 1, \dots, D. \quad (25)$$

Using the previously defined chain rule in (15), it can be shown that the overall error rate with respect to the consequent parameter b_j^i is given according to (17) as follows

$$\begin{aligned} \frac{\partial E}{\partial b_j^i} = & \sum_{p=1}^N \frac{\partial E_p}{\partial b_j^i} \\ = & \sum_{p=1}^N \bar{w}_i \sum_{m=1}^{M_p} \left[\frac{1}{\left(\sum_{h=1}^{M_p} \exp(\alpha(\mathbf{x}_{ph} \cdot \mathbf{b}^i - \mathbf{x}_{pm} \cdot \mathbf{b}^i)) \right)^2} \right. \\ & \times \left[\left(x_{pmj} \sum_{h=1}^{M_p} \exp(\alpha(\mathbf{x}_{ph} \cdot \mathbf{b}^i - \mathbf{x}_{pm} \cdot \mathbf{b}^i)) \right) \right. \\ & \left. \left. - \left(\mathbf{x}_{pm} \cdot \mathbf{b}^i \sum_{h=1}^{M_p} \exp(\alpha(\mathbf{x}_{ph} \cdot \mathbf{b}^i - \mathbf{x}_{pm} \cdot \mathbf{b}^i)) \alpha(x_{phj} - x_{pmj}) \right) \right] \right]. \quad (26) \end{aligned}$$

Hence, the update formula for consequent parameter b_j^i

$$\Delta b_j^i = -\eta \frac{\partial E}{\partial b_j^i}, \quad (27)$$

where η is the same learning rate as in (21)

Equation (27) will be used to update b_j^i either on-line or off-line, depending on the MI-ANFIS implementation.

So far, we have derived all necessary update formulas for the MI-ANFIS premise and consequent parameters. Next, we present our MI-ANFIS basic learning algorithm. It is an iterative algorithm that involves successive updates of the premise and consequent parameters. It is summarized in Algorithm 1.

Algorithm 1 MI-ANFIS Basic Learning Algorithm

Inputs: \mathcal{B} : the set of training bags.

\mathcal{T} : the set of training labels.

M : the number of instances in each bag.

α : the constant used in the “softmax” function.

η : the learning rate.

e : number of epochs.

Outputs: \mathbf{b}^i : the sets of consequent parameters.

\mathbf{c}^i : the set of membership functions’ centers.

σ^i : the set of membership functions’ widths.

Initialize \mathbf{b}^i , \mathbf{c}^i , and σ^i .

repeat

 Update \mathbf{b}^i using (27).

 Update \mathbf{c}^i using (21).

 Update σ^i using (23).

until parameters do not change significantly or number of epochs is exceeded

return \mathbf{b}^i , \mathbf{c}^i , σ^i

IV. EXPERIMENTAL RESULTS

In the following, we report on the experiments conducted to validate the proposed MI-ANFIS. First, we use a synthetic data set to show the potential of MI-ANFIS to learn concepts from ambiguously labeled data. Later, we apply our method to benchmark data sets commonly used in MIL problems and report results.

A. Synthetic Data

We use a simple synthetic dataset to illustrate the potential of using MI-ANFIS to learn concepts from ambiguously labeled data. For this purpose, we generated a synthetic dataset from a distribution of two positive concepts, marked with black and red squares in Figure 2 (the concept points are unknown to MI-ANFIS). From each positive concept we generated 50 bags. We also generated 50 negative bags randomly from non concept regions. Each bag has up to 10 instances. The data is shown in Figure 2. Instances from negative bags are shown as blue letters, and instances from positive bags are shown in red or black letters depending on the underlying concept. Instances from the same bag are displayed using the same letter. In Figure 2, we highlight one bag from Concept 1 by circling all of its instances. As it can be seen, one instance is close to the dense red region (positive concept) while the other instances are scattered around. Positive bags are assigned a labeled of 1, and negative bags are labeled with zeros.

In the following, for the purpose of demonstration we apply only update equations of the premise parameters during the training epochs, and show that the MI-ANFIS Basic Learning Algorithm (Algorithm 1) is capable of identifying positive concepts as well as their corresponding multiple instance fuzzy rules. To initialize the premise parameters, we use the FCM [10] algorithm to partition the instances’ space into 4 clusters². We use the clusters’ centers as initial centers for the Gaussian MFs, and we initialize all standard deviation parameters to a default value of 0.5.

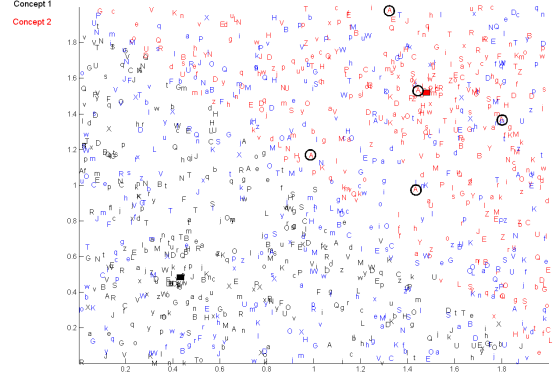


Fig. 2. Instances from positive and negative bags drawn from data that have 2 concepts

The initial fuzzy sets (MFs) of the rules’ premise parts before training are displayed in Figure 3(a). Updated parameters after 20 training epochs are shown in Figure 3(b), and learned fuzzy sets after convergence are shown in Figure 3(c).

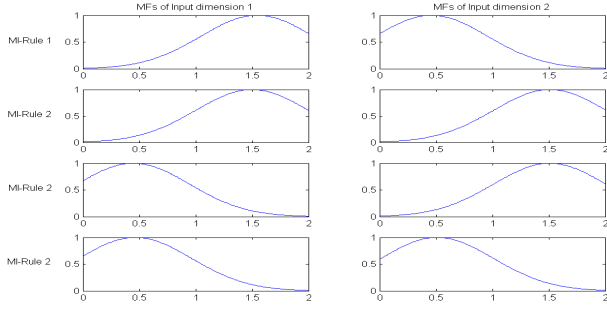
As it can be seen, the algorithm has identified the two true concepts showing that MI-ANFIS can efficiently learn from partially labeled data. More importantly, the system has correctly identified the positive concepts, and at the same time identified irrelevant rules (MI-Rule 1 and MI-Rule 3 marked with red crosses in Figure 3(c)). After training, it is recommended to detect and prune such rules to improve MI-ANFIS testing efficiency. This can be achieved by setting a minimum acceptable fuzzy sets support below which rules containing the set are considered irrelevant.

B. Benchmark Datasets

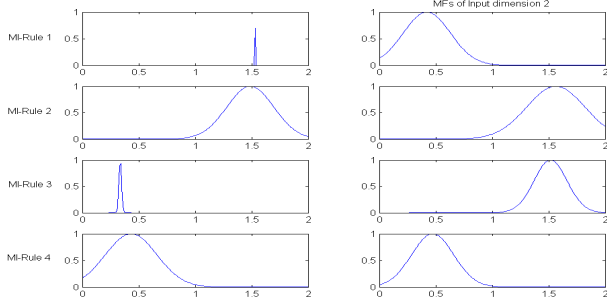
To provide a qualitative evaluation of the proposed MI-ANFIS, we apply it to five benchmark data sets commonly used to evaluate MIL methods. The data sets are namely the MUSK1, MUSK2 [11], and Fox, Tiger, and Elephant from the COREL data set [12]. MUSK1 and MUSK2 data sets consist of descriptions of molecules and the object is to classify whether a molecule smell musky [13]. In these data sets, each bag represents a molecule. Instances in a bag represent the different low-energy conformations of the molecule. Each instance consists of 166 features. MUSK1 has 92 bags, of which 47 are positive, and MUSK2 has 102 bags, of which 39 are positive. The other data sets from COREL: Fox, Tiger, and Elephant, classify whether an image contains the corresponding animal. Each data set consists of 200 images (bags): 100 positive images containing the target animal and 100 negative images containing other animals. Each image is represented as a set of patches (instances) and each patch is in turn represented by 230 features describing color, texture and shape information. Table I summarizes the characteristics of the five data sets. It is to be noted that for each benchmark data set, PCA was applied to reduce the dimensionality of the features in order to speedup MI-ANFIS training and increase the interpretability of the generated multiple instance fuzzy rules.

For all experiments, we construct a zero-order MI-ANFIS

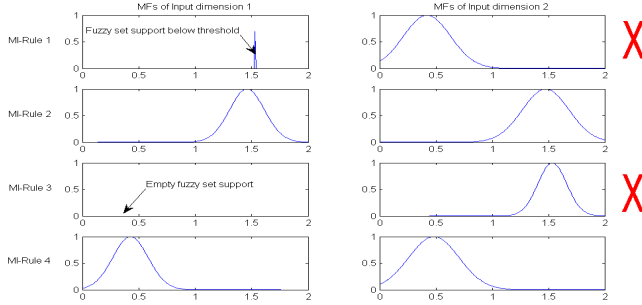
²A grid or manual partitioning could also be used



(a) Initial MFs before starting the training process.



(b) Input MFs during MI-ANFIS training (Epoch number 20).



(c) Learned MFs after convergence of MI-ANFIS training algorithm. Rules marked with red crosses are considered vanished and are pruned. Remaining rules (MI-Rule 2 and MI-Rule 4) correctly describe the positive concepts of the dataset

Fig. 3. Input MFs before, during, and after MI-ANFIS training.

TABLE I. BENCHMARK DATA SETS

Data set	dim.(PCA)	No. Bags	Positive	Negative	No.Instances
MUSK1	166(25)	92	47	45	2 → 40
MUSK2	166(25)	102	39	63	1 → 1044
Fox	230(10)	200	100	100	2 → 13
Tiger	230(10)	200	100	100	1 → 13
Elephant	230(10)	200	100	100	2 → 13

(constant consequent parameters) having 15 multiple instance rules, and employing Gaussian MFs to describe the input fuzzy sets. For initialization, we use the FCM algorithm to cluster the instances of the positive bags into 15 clusters, and we initialize MFs' centers as the clusters centers. Table II summarizes all parameters used in training the MI-ANFIS. We note that the reason behind using large standard deviations for MUSK1, MUSK2, and FOX datasets is to allow the

TABLE II. MI-ANFIS TRAINING PARAMETERS

Parameter	MUSK1	MUSK2	FOX	Tiger	Elephant
No. of MI Rules	15	15	15	15	15
No. of Inputs	25	25	10	10	10
MF's σ	100	100	100	10	10
Output parameters	1s	1s	1s	1s	1s
Softmax's α	1	1	1	1	1
Learning rate	0.1	0.1	0.1	0.1	0.1

TABLE III. COMPARISON OF MI-ANFIS PREDICTION ACCURACY (IN PERCENT) TO OTHER METHODS ON THE BENCHMARK DATA SETS. RESULTS FOR 3 TOP PERFORMING METHODS ARE SHOWN IN BOLD FONT.

Algorithms	MUSK1	MUSK2	Fox	Tiger	Elephant
MI-ANFIS	93.49 ± 0.76	90.58 ± 1.31	66.4 ± 2.77	84.5 ± 0.61	86.97 ± 1.10
MILES [14]	86.3	87.7	N/A	N/A	N/A
APR [11]	92.4	89.2	N/A	N/A	N/A
DD [8]	88.9	82.5	N/A	N/A	N/A
DD-SVM [15]	85.8	91.3	N/A	N/A	N/A
EM-DD [16]	84.8	84.9	56.1	72.1	78.3
Citation-KNN [17]	92.4	86.3	N/A	N/A	N/A
MI-SVM [12]	77.9	84.3	57.8	84.0	81.4
mi-SVM [12]	87.4	83.6	58.2	78.4	82.2
MI-NN [18]	88.0	82.0	N/A	N/A	N/A
Bagging-APR [19]	92.8	93.1	N/A	N/A	N/A
RBF-MIP [20]	91.3 ± 1.6	90.1 ± 1.7	N/A	N/A	N/A
BP-MIP [21]	83.7	80.4	N/A	N/A	N/A
RBF-Bag-Unit [22]	90.3	86.6	N/A	N/A	N/A
MI-kernel [23]	88.0	89.3	60.3	84.2	84.3
PPPM-kernel [24]	95.6	81.2	60.3	80.2	82.4
MIGraph [23]	90.0	90.0	61.2	81.9	85.1
miGraph [23]	88.9	90.3	61.6	86.0	86.8
ALP-SVM [25]	86.3	86.2	66.0	86.0	83.5
MIForest [26]	85.0	82.0	64.0	82.0	84.0
Naive-ANFIS	67.82 ± 4.04	79.43 ± 5.04	N/A	N/A	N/A

initial rules to cover the entirety of the input space.

After initialization, we run MI-ANFIS basic learning algorithm (Algorithm 1) to jointly learn a fuzzy description of positive concepts as well as the optimal multiple instance rules' output.

Table III shows the performance of the proposed algorithm on the benchmark data sets. MI-ANFIS was trained and tested using ten fold cross validation. The performance is reported in terms of prediction accuracy (% of correct \pm standard deviation).

To show the advantage of using MI-ANFIS over the traditional ANFIS we compare its performance to the later on the benchmark data sets. Given that ANFIS cannot learn from ambiguously labeled data, for sake of comparison, we consider the naive MIL assumption where all instances from positive bags are considered positive and all instances from negative bags are considered negative. We refer to this implementation as Naive-ANFIS. An empirical comparison with other MIL methods is also reported.

Overall, MI-ANFIS achieved state of the art performances. On all tested data sets, MI-ANFIS ranked consistently among

the top three. For MUSK1, PPM-kernel [24] performed the best (95.6%), but did not perform as well for the other sets. For MUSK2 Bagging-APR [19] achieved the best accuracy, as reported by [14], Bagging-APR excellent performance is credited to the use of an ensemble scheme to the base learner APR [11]. MI-ANFIS achieved the best average performance for the Fox and Elephant data sets, and second best performance after the miGraph [23] and ALP-SVM [25] methods for the Tiger data set. It is clear from Table III that Naive-ANFIS performed the worse, this is basically due to the naive MIL assumption. In cases where more information about instances is available, such information could be used to relax the naive assumption by assigning better labels at the instances' level, and could lead to better ANFIS (standard) performance.

C. Discussion

Fuzzy logic is powerful at modeling knowledge uncertainty and measurements imprecision [27]. More generally, it is one of the best frameworks to model vagueness. However, in addition to uncertainty and imprecision, there is a third vagueness concept that fuzzy logic does not address quite well. This vagueness concept is due to the ambiguity that arises when the data have multiple forms of expression, this is the case for multiple instance problems. MI-ANFIS deals with ambiguity by introducing the novel concept of truth instances: when carrying reasoning using a bag of instances at Layer 2 (Figure 1), a proposition will not only have one degree of truth, it will have multiple degrees of truth (r_{ij}), we call truth instances. Thus, effectively encoding the third vagueness component of ambiguity and increasing the expressive power of traditional fuzzy logic.

Learning positive concepts from ambiguously labeled data has been the core task of various MIL algorithms (e.g. Diverse Density [8]). MI-ANFIS has proven that it can learn positive concepts effectively while jointly providing a fuzzy representation of such regions. The fuzzy representation is combined into meaningful and simple multiple instance rules that can be easily visualized and interpreted. The fuzzy representation also offers the advantage of robustness against noise points that might happen to be close to positive concepts without being necessarily positive. Thus, lowering the amount of false positives. MI-ANFIS is fully independent. It does not require positive concepts to be learned using a different algorithm (e.g. Diverse Density [8]), or based on intuition. Moreover, MI-ANFIS does not rely on any traditional MIL algorithms and can learn its rule base from data.

V. CONCLUSIONS

In this paper, we presented MI-ANFIS, an novel neuro-fuzzy architecture that extends the standard Adaptive Neuro-Fuzzy Inference System (ANFIS) to reason with bags of instances in order to solve multiple instance learning problems. We developed a BackPropagation learning algorithm using a thoroughly and abstract mathematical formulation and showed that the proposed system is capable of learning meaningful concepts from ambiguously labeled data. We reported on the performance of the proposed algorithm using a synthetic and five benchmark data sets, in different scenario MI-ANFIS showed promising results.

In future work, we intend to develop a hybrid learning algorithm that combines a gradient method and a least squares

estimator (LSE), in order to speedup MI-ANFIS training. We will also report on the complexity of the developed training algorithms.

ACKNOWLEDGMENT

This work was supported in part by U.S. Army Research Office Grants Number W911NF-13-1-0066 and W911NF-14-1-0589. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, or the U.S. Government.

REFERENCES

- [1] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]," *Automatic Control, IEEE Transactions on*, vol. 42, no. 10, pp. 1482–1484, 1997.
- [2] Y. Chai, L. Jia, and Z. Zhang, "Mamdani model based adaptive neural fuzzy inference system and its application," *International Journal of Computational Intelligence*, vol. 5, no. 1, pp. 22–29, 2009.
- [3] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *Computers, IEEE Transactions on*, vol. C-26, no. 12, pp. 1182–1191, Dec 1977.
- [4] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-15, no. 1, pp. 116–132, 1985. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6313399>
- [5] O. Maron, "Learning from ambiguity," Ph.D. dissertation, Massachusetts Institute of Technology, 1998.
- [6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 12, pp. 31 – 71, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370296000343>
- [7] A. B. Khalifa and H. Frigui, "Fusion of multiple landmine detection algorithms using an adaptive neuro fuzzy inference system," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*. IEEE, 2014, pp. 3148–3151.
- [8] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, ser. NIPS '97. Cambridge, MA, USA: MIT Press, 1998, pp. 570–576. [Online]. Available: <http://dl.acm.org/citation.cfm?id=302528.302753>
- [9] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 3, pp. 665–685, 1993.
- [10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997.
- [12] S. Andrews, I. Tsochantaris, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2002, pp. 561–568.
- [13] Y. Li, D. M. Tax, R. P. Duin, and M. Loog, "Multiple-instance learning as a classifier combining problem," *Pattern Recognition*, vol. 46, no. 3, pp. 865–874, 2013.
- [14] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 1931–1947, 2006.
- [15] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *The Journal of Machine Learning Research*, vol. 5, pp. 913–939, 2004.
- [16] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *Advances in neural information processing systems*, 2001, pp. 1073–1080.

- [17] J. Wang and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach," 2000.
- [18] J. Ramon and L. De Raedt, "Multi instance neural networks," 2000.
- [19] Z.-H. Zhou and M.-L. Zhang, "Ensembles of multi-instance learners," in *Machine Learning: ECML 2003*. Springer, 2003, pp. 492–502.
- [20] M.-L. Zhang and Z.-H. Zhou, "Adapting rbf neural networks to multi-instance learning," *Neural Processing Letters*, vol. 23, no. 1, pp. 1–26, 2006.
- [21] Z.-H. Zhou and M.-L. Zhang, "Neural networks for multi-instance learning," *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pp. 455–459, 2002.
- [22] A. Bouchachia, "Multiple instance learning with radial basis function neural networks," in *Neural Information Processing*. Springer, 2002, Conference Proceedings, pp. 440–445.
- [23] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-iid samples," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 1249–1256.
- [24] H.-Y. Wang, Q. Yang, and H. Zha, "Adaptive p-posterior mixture-model kernels for multiple instance learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1136–1143.
- [25] P. V. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning," in *International conference on artificial intelligence and statistics*, 2007, pp. 123–130.
- [26] C. Leistner, A. Saffari, and H. Bischof, "Miforests: multiple-instance learning with randomized trees," in *Computer Vision—ECCV 2010*. Springer, 2010, pp. 29–42.
- [27] L. A. Zadeh, "Is there a need for fuzzy logic?" *Information Sciences*, vol. 178, no. 13, pp. 2751–2779, 2008.